NPS52-84-007

NAVAL POSTGRADUATE SCHOOL

Monterey, California

A SIMPLE PROOF OF A GENERALIZED
CHURCH-ROSSER THEOREM

B. J. MacLennan

June 1984

Prepared for:

Chief of Naval Research
rlington, VA 22217

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Commodore R. H. Shumaker
Superintendent

D. A. Schrady
Provost

The work reported herein was supported by the Office of Naval Research under contract number N00014-84-WR-24087.

This report was prepared by:

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NPS52-84-007	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Simple Proof of a Generalized Church-Rosser Theorem		5. TYPE OF REPORT & PERIOD COVERED Technical Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) B. J. MacLennan		8. CONTRACT OR GRANT NUMBER(s) N00014-84-WR-24087
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61153N; RR014-08-01 N0001484WR24087
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Arlington, Virginia 22217		12. REPORT DATE June 1984
		13. NUMBER OF PAGES 14
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release, distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Church-Rosser Theorem, abstract calculus, tree transformation, term rewriting, functional programming, logic programming, equational programming, production systems, tree, nondeterministic algorithm, normal form, lambda calculus

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Abstract calculi (tree transformation systems, term rewriting systems) express computational processes by transformation rules operating on abstract structures (trees). They have applications to functional programming, logic programming, equational programming, productions systems and language processors.

(continued)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102- LF- 014- 6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

We present a proof of the Church-Rosser Theorem for a wide, useful class of abstract calculi. This theorem implies that terminating reductions always yield a unique reduced form in these calculi, which has the practical result that transformation rules can be safely applied in any order, or even in parallel. Although this result has previously been established for certain classes of abstract calculi, our proof is much simpler than previous proofs because it is an adaption of Rosser's new (1982) proof of the Church-Rosser Theorem for the lambda calculus.

A Simple Proof of a Generalized Church-Rosser Theorem

B. J. MacLennan

Computer Science Department

Naval Postgraduate School

Monterey, CA 93943

Abstract

Abstract calculi (tree transformation systems, term rewriting systems) express computational processes by transformation rules operating on abstract structures (trees). They have applications to functional programming, logic programming, equational programming, productions systems and language processors.

We present a proof of the Church-Rosser Theorem for a wide, useful class of abstract calculi. This theorem implies that terminating reductions always yield a unique reduced form in these calculi, which has the practical result that transformation rules can be safely applied in any order, or even in parallel. Athough this result has previously been established for certain classes of abstract calculi, our proof is much simpler than previous proofs because it is an adaptation of Rosser's new (1982) proof of the Church-Rosser Theorem for the lambda calculus.

1. Introduction

A calculus has the *Church-Rosser Property* if any terminating reduction of a formula of the calculus always yields the same reduced form. The *Church-Rosser Theorem*, originally proved in 1936 [5], implies that the lambda calculus has the Church-Rosser Property. In this paper we prove that the Church-Rosser Property holds for a certain wide, useful class of abstract calculi (also known as tree transformation systems and term rewriting systems).

Abstract calculi are increasingly relevant, due to their applications in functional programming [1, 3, 10], logic programming [6, 8, 17], equational programming [4, 9,

14, 15, 16] and production systems [19, 20]. The Church-Rosser properties of abstract calculi have been previously investigated [21]. Our proof differs from its predecessors in making use of an adaptation of a new, simplified proof of the Church-Rosser Theorem first presented by Rosser in 1982 [22]. Rosser's new proof is a vast improvement on its predecessors [2, 5, 7, 11, 12, 13, 18, 21, 23], and for the first time makes it feasible to prove the Church-Rosser Theorem in classroom situations.

2. Definitions and Notation

2.1 Syntax

Atoms, Formulas and Lists: We assume a denumerably infinite universe of *atoms*. A *formula* can either be *atomic*, which means it is an atom, or *nonatomic*, which means it is an expression of the form $(F_1\ F_2\ \cdots\ F_n)$, in which the F_i are formulas and $n \geq 0$. Formulas are required to be finite. Nonatomic formulas are often called *lists*. A list of *depth one* is a list all of whose elements are atoms. A list of depth $n > 1$ is a list containing at least one list of depth $n-1$.

Abstract Calculus: An *abstract calculus* is a pair $<\Sigma, R>$ in which Σ is a set of atoms, called the *reserved symbols* of the calculus, and R is a set of *transformation rules* satisfying the following definition.

Transformation Rules. Each transformation rules has the form $A \Rightarrow S$ in which A and S are formulas. A is called the *analysis part* or *pattern* of the rule, and S is called the *synthesis part* or *template* of the rule. Atoms in A are called *pattern constants* if they are reserved symbols of the calculus, and are called *pattern variables* if they are not. We do not permit rules to be *universally applicable*, a notion defined next.

Universally Applicable Rules: A rule is called *universally applicable* if its analysis part is atomic but not a reserved symbol. That is, it is a rule of the form $x \Rightarrow S$ in which x is a pattern variable. We discuss later the reasons we do not permit universally applicable rules. A rule is *nonuniversally applicable* if its analysis part is either a list

or a reserved symbol.

2.2 Semantics

Substitution: If x is an atom and X and V are formulas, then $X[x=V]$ refers to the result of substituting V for all occurrences of x. This operation can be defined recursively as follows:

$$(X_1 X_2 \cdots X_n)[x=V] \Rightarrow (X_1[x=V]\, X_2[x=V] \cdots X_n[x=V])$$

$$x[x=V] \Rightarrow V$$

$$y[x=V] \Rightarrow y, \text{ if } y \text{ is atomic and } x \neq y$$

If the atoms $x_1, x_2, \ldots, x_n$ are all distinct, then we write $X[x_1=V_1, x_2=V_2, \ldots, x_n=V_n]$ for $X[x_1=V_1][x_2=V_2] \cdots [x_n=V_n]$.

Notice that substitution satisfies the following property:

$$M[x_1=X_1, \ldots, x_m=X_n][y_1=Y_1, \ldots, y_n=Y_n]$$
$$= M[x_1 = X_1[y_1=Y_1, \ldots, y_n=Y_n], \ldots, x_m = X_m[y_1=Y_1, \ldots, y_n=Y_n]]$$

Matching and Transformation: We say that a formula X *matches* the analysis part of a rule $A \Rightarrow S$ if there are formulas $V_1, \ldots, V_n$ such that $X = A[v_1=V_1, \ldots, v_n=V_n]$, where $v_1, \ldots, v_n$ are the pattern variables of A. If this is the case then we can *apply* the rule to X to yield $S[v_1=V_1, \ldots, v_n=V_n]$.

Subformulas: A formula X is called a *subformula* of a formula Y if there is some formula M and atom x in M such that $Y = M[x=X]$. X is called a *proper subformula* of Y if this M is nonatomic.

Reduction: We say that a transformation rule *reduces* a formula Y to a formula Z if that rule can be applied to a subformula of Y. That is, if $Y = M[x=X]$ and the transformation rule when applied to X yields W, then we say the transformation rule reduces Y to $Z = M[x=W]$. More precisely, if $A \Rightarrow S$ is the rule, $Y = M[x=X]$, and $X = A[v_1=V_1, \ldots, v_n=V_n]$, then $Z = M[x=W]$, where $W = S[v_1=V_1, \ldots, v_n=V_n]$. A *reduction* is a finite sequence of formulas $X_1, \ldots, X_n$, such that for each X_i there is a

transformation rule that reduces X_i to X_{i+1}. We say that a formula X *can be reduced* to a formula Y when either $X=Y$ or there is a reduction $X_1, \ldots, X_n$ such that $X=X_1$ and $X_n=Y$. A formula is in *reduced form* when none of the transformation rules can be applied to it.

Walks: We call a reduction from X to Y a *walk* and write 'X **walk** Y' when the reduction satisfies the *walk restriction*. Informally the walk restriction says that the reduction must proceed from the inside out. That is, we can never apply a transformation rule to a subformula W of X, and then later in the reduction sequence transform a subformula (proper or otherwise) of W. Of course, the walk restriction does not constrain the order in which rules are applied to nonoverlapping subformulas.

Diamond Property: A relation R has the *diamond property* if $X\ R\ X_1$ and $X\ R\ X_2$ implies that there exists an X' such that $X_1\ R\ X'$ and $X_2\ R\ X'$.

2.3 Restrictions

Ambiguous Rules: A set of transformation rules is *ambiguous* if there exist distinct rules $A \Rightarrow S$ and $A' \Rightarrow S'$ in the set such that there is a formula X that matches both A and A'.

Conflicting Rules: Two transformation rules $A \Rightarrow S$ and $A' \Rightarrow S'$ (not necessarily distinct) are *conflicting* if there is a formula that matches both A and a proper subpart of A'. That is, $A \Rightarrow S$ and $A' \Rightarrow S'$ are conflicting if (1) A' can be written as $M[x=B]$, where M is nonatomic, and (2) there are formulas $V_1, \ldots, V_n$ such that $B[v_1=V_1, \ldots, v_n=V_n] = A[v_1=V_1, \ldots, v_n=V_n]$, where $v_1, \ldots, v_n$ are the pattern variables that appear in B or A or both. Note that a transformation rule can conflict with itself. A set of transformation rules is *nonconflicting* if none of the rules conflict with each other.

Atomic Rules: A transformation rule is *atomic* if its analysis part is an atom. Since we require rules to be nonuniversally applicable, the only permitted atomic rules are those whose analysis part is a reserved symbol. A transformation rule is *nonatomic* if it is not atomic. A set of transformation rules is *anti-atomic* if all its rules are

nonatomic.

Universally Applicable Rules: We can now explain why we do not permit universally applicable rules. The first reason is that such rules are *inherently ambiguous* if they are combined with any other rules. This is because the analysis part of a universally applicable rule matches any formula (hence, their name). Therefore, the only unambiguous calculus containing such a rule is the calculus that has only that rule. The second reason is that these rules are not very useful: Since these rules are always applicable, a calculus containing them has no terminating reductions and no reduced forms.

3. Walks Have the Diamond Property

Theorem: Walks have the diamond property in an abstract calculus with unambiguous, nonconflicting, anti-atomic rules .

Proof: We must show that if X **walk** X_1 and X **walk** X_2, then there is an X' such that X_1 **walk** X' and X_2 **walk** X'. The proof is by induction on the size of X.

Suppose X has size one, that is, X is an atom. Since the rules are anti-atomic there are no transformation rules applicable to X. Hence, $X = X_1 = X_2 = X'$ satisfies the theorem. This establishes the base of the recursion.

Now suppose that X is nonatomic. There are four cases depending on whether or not the top level of X is transformed in each of the walks, X **walk** X_1 and X **walk** X_2.

Case 1: Suppose that the top level is transformed in neither walk. Now, since X is nonatomic, it can be written in the form $M[x_1=U_1, \ldots, x_n=U_n]$, where M is a list of atoms (i.e., list of depth one). Since the top level of X is not transformed, X_1 can be written in the form $M[x_1=V_1, \ldots, x_n=V_n]$, and X_2 can be written in the form $M[x_1=W_1, \ldots, x_n=W_n]$. Since X **walk** X_1 we know U_i **walk** V_i, for $1 \le i \le n$. Similarly, since X **walk** X_2, we know U_i **walk** W_i, for $1 \le i \le n$. Since the U_i are proper subparts of X, they are smaller than X, so we can apply the inductive hypothesis and conclude that

there are Y_i such that V_i **walk** Y_i and W_i **walk** Y_i $(1 \le i \le n)$. We let $X' = M[x_1 = Y_1, \ldots, x_n = Y_n]$.

We still must show that X_1 **walk** X' and X_2 **walk** X'. But this is easy: since the formulas V_i are nonoverlapping, walking them to the Y_i cannot violate the walk restriction. Hence, walking the V_i individually to the Y_i will walk X_1 to X'. Similarly, we can walk X_2 to X'.

Case 2: Here we assume that the top level of X has been transformed in the walk from X to X_1, but not in the walk from X to X_2. However, to avoid violating the walk restriction, the top level transformation must have been the last step in the walk from X to X_1. Hence the last step takes a formula Z into X_1 by an application of a transformation rule $A \Rightarrow S$, in which $Z = A[v_1 = M_1, \ldots, v_m = M_m]$ and $X_1 = S[v_1 = M_1, \ldots, v_m = M_m]$ Since this last step is the only one to transform the top level of X, by the same reasoning as in Case 1 we know there is an M of depth one such that

$$X = M[x_1 = U_1, \ldots, x_n = U_n]$$
$$Z = M[x_1 = V_1, \ldots, x_n = V_n]$$
$$X_2 = M[x_1 = W_1, \ldots, x_n = W_n]$$

U_i **walk** V_i, U_i **walk** W_i, for $1 \le i \le n$

Now, since A matches the top level of X, A can be written in the form $M[x_1 = A_1, \ldots, x_n = A_n]$. Applying properties of the substitution operator Z can be written in the form

$$\begin{aligned} Z &= A[v_1 = M_1, \ldots, v_m = M_m] \\ &= M[x_1 = A_1, \ldots, x_n = A_n][v_1 = M_1, \ldots, v_m = M_m] \\ &= M[x_1 = A_1[v_1 = M_1, \ldots, v_m = M_m], \ldots, x_n = A_n[v_1 = M_1, \ldots, v_m = M_m]] \end{aligned}$$

Hence, $V_i = A_i[v_1 = M_1, \ldots, v_m = M_m]$, for $1 \le i \le n$.

By the inductive hypothesis we know there are Y_i such that V_i **walk** Y_i and W_i **walk** Y_i, $1 \le i \le n$. We claim that Y_i must have the form $A_i[v_1 = Q_1, \ldots, v_m = Q_m]$, for

some Q_i, $1 \le i \le n$. If A_i is the variable v_k then this is trivially true; let $Q_k = Y_k$. If A_i is nonatomic, then the only way there could fail to be such Q_j would be if the walk from V_i to Y_i altered the top level structure of V_i, that is, eliminated the A_i structure. But this would violate the nonconflicting restriction on rules, since it would require a transformation rule with A_i on the top level of the analysis part. Since A_i is a proper subpart of the analysis part of the rule $A \Rightarrow S$, we would have conflicting rules. Thus we know $Y_i = A_i[v_1=Q_1, \ldots, v_m=Q_m]$. We will take our X' to be $S[v_1=Q_1, \ldots, v_m=Q_m]$. It remains to show that there are walks from X_1 and X_2 to this X'.

In the case of X_1 this is trivial. Note that since we have V_i **walk** Y_i, we also have

$$A_i[v_1=M_1, \ldots, v_m=M_m] \textbf{ walk } A_i[v_1=Q_1, \ldots, v_m=Q_m]$$

We can separate out from this walk the individual steps that apply to each of the M_j. Thus we have M_j **walk** Q_j, for $1 \le j \le m$. Since $X_1 = S[v_1=M_1, \ldots, v_m=M_m]$, we can walk each of the M_j in X_1 individually to the corresponding Q_j. Since the M_j do not overlap, the catenation of the individual walks is itself a walk, so we have a walk from X_1 to $S[v_1=Q_1, \ldots, v_m=Q_m]$. But the latter is just X', so we have X_1 **walk** X'.

We now consider the walk from X_2 to X'. Since $X_2 = M[x_1=W_1, \ldots, x_n=W_n]$, we can walk each of the W_i individually to Y_i yielding $Y = M[x_1=Y_1, \ldots, x_n=Y_n]$. The catenation of these individual walks is a walk, because the W_i are nonoverlapping. Hence there is a walk from X_2 to Y.

Now, since $Y_i = A_i[v_1=Q_1, \ldots, v_m=Q_m]$, we can rewrite Y as follows:

$$\begin{aligned}
Y &= M[x_1=Y_1, \ldots, x_n=Y_n] \\
&= M[x_1 = A_1[v_1=Q_1, \ldots, v_m=Q_m], \ldots, x_n = A_n[v_1=Q_1, \ldots, v_m=Q_m]] \\
&= M[x_1=A_1, \ldots, x_n=A_n][v_1=Q_1, \ldots, v_m=Q_m] \\
&= A[v_1=Q_1, \ldots, v_m=Q_m]
\end{aligned}$$

The latter matches the left-hand side of the transformation rule $A \Rightarrow S$, so we can apply this rule to yield $S[v_1=Q_1, \ldots, v_n=Q_n]$, which is X'. The catenation of the walk from

X_2 to Y with the transformation from Y to X' is a walk, since this transformation applies at the top level, and no previous rules applied at that level. Thus we have a walk from X_2 to X'. This complete Case 2.

Case 3: This case is the opposite of Case 2; the top level is transformed in the walk from X to X_2 but not in that from X to X_1. The proof is exactly analogous to Case 2.

Case 4: In this case the top level is transformed in both the walks, from X to X_1 and from X to X_2. By the same reasoning as in Case 2, the last step in each of these walks must be a top level transformation. Thus we have a walk from X to a formula Y such that the application of a rule $A \Rightarrow S$ to the top level of Y yields X_1. Similarly we have a walk from X to a formula Z such that the application of a rule $A' \Rightarrow S'$ to the top level of Z yields X_2. Since neither the walk from X to Y nor from X to Z can have altered the top level structure of X, we can write X, Y and Z as substitutions into a depth one list M:

$$
\begin{aligned}
X &= M[x_1 = U_1, \ldots, x_n = U_n] \\
Y &= M[x_1 = V_1, \ldots, x_n = V_n] \\
Z &= M[x_1 = W_1, \ldots, x_n = W_n]
\end{aligned}
$$

We know that U_i **walk** V_i and U_i **walk** W_i, so by the inductive hypothesis there are Y_i such that V_i **walk** Y_i and W_i **walk** Y_i, all for $1 \le i \le n$. Let $Q = M[x_1 = Y_1, \ldots, x_n = Y_n]$.

We know that the top level of Y matches the left-hand side of $A \Rightarrow S$, and that the top level of Z matches the left-hand side of $A' \Rightarrow S'$. We intend to show that the top level of Q matches the left-hand side of both rules. From this, and the hypothesis that the rules are unambiguous, we will be able to conclude that the rules $A \Rightarrow S$ and $A' \Rightarrow S'$ are the same rule. The remainder of the proof then follows easily.

We now show that the top level of Q matches A. By the same reasoning as in Case 2 we know that there are A_i such that $Y_i = A_i[v_1 = Q_1, \ldots, v_m = Q_m]$, where $v_1, \ldots, v_m$ are the pattern variables of A and $A = M[x_1 = A_1, \ldots, x_n = A_n]$. Hence we can rewrite Q as follows:

$$
\begin{aligned}
Q &= M[x_1 = Y_1, \ldots, x_n = Y_n] \\
&= M[x_1 = A_1[v_1 = Q_1, \ldots, v_m = Q_m], \ldots, A_n[v_1 = Q_1, \ldots, v_m = Q_m]] \\
&= M[x_1 = A_1, \ldots, x_n = A_n][v_1 = Q_1, \ldots, v_m = Q_m] \\
&= A[v_1 = Q_1, \ldots, v_m = Q_m]
\end{aligned}
$$

Thus the top level of Q matches A. Exactly analogous reasoning shows that the top level of Q match A'. Since our rules are unambiguous, we conclude that $A = A'$ and $S = S'$.

We now define X' to be $S[v_1 = Q_1, \ldots, v_m = Q_m]$. It remains to show that X_1 **walk** X' and X_2 **walk** X'. The reasoning is analogous to that used in Case 2. Since Y_i **walk** Y_i, $1 \le i \le n$, we know

$$A_i[v_1 = M_1, \ldots, v_m = M_m] \ \textbf{walk} \ A_i[v_1 = Q_1, \ldots, v_m = Q_m]$$

Hence we know M_j **walk** Q_j, for $1 \le j \le m$. Since $X_1 = S[v_1 = M_1, \ldots, v_m = M_m]$, we can walk each of the M_j individually to Q_j to yield $S[v_1 = Q_1, \ldots, v_m = Q_m]$, which is X'. Since the M_j are nonoverlapping, the catenation of these walks is a walk from X_1 to X'. Exactly the same reasoning applies to show a walk from X_2 to X'. This completes the proof of Case 4, and hence the proof of the theorem. *Q.E.D.*

Reduction has the Diamond Property

The following lemma shows that finite sequences of walks have the diamond property. We use the notation X_1 **walk*** X_n to mean that there is a sequence of walks from X_1 to X_n, that is,

$$X_1 \ \textbf{walk} \ X_2 \ \textbf{walk} \ \cdots \ \textbf{walk} \ X_{n-1} \ \textbf{walk} \ X_n$$

We permit $n=1$, that is, $X_1 = X_n$.

Lemma: If X **walk*** Y and X **walk*** **walk*** Z then there is an X' such that Y **walk*** X' and Z **walk*** X'.

Proof: We prove a slightly stronger result: if there are m walks in the reduction of X

to Y and n in the reduction of X to Z, then there is an X' that can be reached by n walks from Y and m walks from Z.

The proof is by induction on the maximum number of walks in the reductions from X to Y and X to Z. If there are no walks, then $X = Y = Z$ and $m=n=0$ so we can take $X' = X$. This establishes the base of the induction.

Now suppose that there are m walks in the reduction of X to Y and n walks in the reduction of X to Z:

X **walk** Y_1 **walk** Y_2 **walk** $\cdots$ **walk** Y_{m-1} **walk** $Y_m = Y$

X **walk** Z_1 **walk** Z_2 **walk** $\cdots$ **walk** Z_{n-1} **walk** $Z_n = Z$

Case 1: Suppose $m>n$. Then we can apply the inductive hypothesis to the reductions from X to Y_{m-1} and X to Z, since these both contain less than m walks. Hence there is a W such that Y_{m-1} reaches W in n walks, and Z reaches W in $m-1$ walks.

Since Y_{m-1} reaches Y by one walk and W by n walks, and $n<m$ (by supposition), we can apply the inductive hypothesis again and conclude that there is an X' such that Y reaches X' by n walks and W reaches X' by one walk. Hence we have found an X' such that Y reaches X' in n walks and Z reaches X' in m walks (since $m-1$ take Z to W and one more takes W to X'). This completes Case 1.

Case 2: Suppose $m=n$. Apply the inductive hypothesis to the reductions from X to Y_{m-1} and from X to Z_{m-1}. Thus there is a W reachable from both Y_{m-1} and Z_{m-1} in $m-1$ walks. Apply the inductive hypothesis to the reductions from Y_{m-1} to Y and W to get a U reachable from Y in $m-1$ walks and from W in one walk. Similarly there is a V reachable from W in one walk and from Z in $m-1$ walks. Apply the inductive hypothesis one last time to the reductions from W to U and V to get an X' reachable from both U and V by one walk. Now, since Y reaches U by $m-1$ walks, and U reaches X' by one walk, Y reaches X' by m walks. Similarly, Z reaches X' by m walks. This completes Case 2.

Case 3: Suppose $m<n$. This is proved exactly like Case 1. Hence the lemma is proved. *Q.E.D.*

Lemma: Reduction has the diamond property in an abstract calculus with unambiguous, nonconflicting, anti-atomic rules.

Proof: Note that a single application of a transformation rule constitutes a legal walk (it takes at least two reduction steps to violate the walk restriction). Since finite sequences of walks have the diamond property, so do finite sequences of rule applications. But a reduction is just a finite sequence of rule applications. *Q.E.D.*

Theorem (Generalized Church-Rosser): Reductions have the diamond property in an abstract calculus with unambiguous, nonconflicting rules.

Proof: This result differs from the previous in that we do not require the rules to be anti-atomic. It was convenient to restrict our attention to anti-atomic sets of rules when proving that walks have the diamond property.

We define a one-to-one correspondence, called *depth-raising*, between formulas as follows. Depth-raising an atomic formula a yields the depth-one list (a). Depth-raising a list $(F_1 \, F_2 \, \cdots \, F_n)$ yields the list $((\overline{F_1} \, \overline{F_2} \, \cdots \, \overline{F_n}))$, in which each $\overline{F_i}$ is the result of depth-raising the corresponding F_i. The effect of depth-raising is to double the number of parentheses surrounding lists, and to place single parentheses around atoms. For example, depth-raising (+ (**suc** 2) 3) yields

(((+) (((**suc**) (2))) (3)))

Depth-raising sets up a one-to-one correspondence between formulas and depth-raised formulas: for each formula there is exactly one depth-raised formula, and for each depth-raised formula there is exactly one formula. The crucial point, of course, is that depth-raised formulas are nonatomic.

We depth-raise an abstract calculus by depth-raising the analysis and synthesis parts of each of its transformation rules. Clearly a formula X matches the analysis

part of one of the original rules if and only if the depth-raised formula $\bar{X}$ matches the analysis part of the corresponding depth-raised analysis part. In particular, an atomic formula a matches the analysis part of an atomic (but nonuniversally applicable) rule $b \Rightarrow S$ if and only if the depth-raised formula (a) matches the depth-raised rule $(b) \Rightarrow S$.

Similarly, one of the original rules transforms a formula X into a formula Y if and only if the depth-raised rule transforms $\bar{X}$ into $\bar{Y}$. It is also clear that a depth-raised transformation rule will always yield a depth-raised formula.

These observations allow us to prove that reductions have the diamond property, even in the presence of atomic rules. Suppose we have an abstract calculus whose rules are unambiguous and nonconflicting, but may be atomic. Let X, X_1 and X_2 be any three formulas such that X reduces to X_1 and X_2. Now consider the corresponding depth-raised calculus and the corresponding formulas $\bar{X}$, $\overline{X_1}$ and $\overline{X_2}$. Clearly $\bar{X}$ reduces to $\overline{X_1}$ and $\overline{X_2}$ in the depth-raised calculus. Since the depth-raised calculus is anti-atomic, we know reductions in it have the diamond property. Hence there is a depth-raised formula $\overline{X'}$ such that both $\overline{X_1}$ and $\overline{X_2}$ reduce to $\overline{X'}$ in this calculus. Therefore we know there is a depth-lowering X' of $\overline{X'}$ such that X_1 and X_2 reduce to X' in the original calculus. Hence reduction also has the diamond property in the original calculus. *Q.E.D.*

Corollary: Reduced forms are unique in an abstract calculus with unambiguous, nonconflicting rules.

Proof: Suppose that we have a reduction from X to Y and a reduction from X to Z and that both Y and Z are in reduced form. Since reduction has the diamond property we know there is a W such that Y reduces to W and Z reduces to W. But, since Y is in reduced form, no rules are applicable to it, so $Y=W$. Similarly, $Z=W$. Hence $Y=Z$ and we see that reduced forms are unique. *Q.E.D.*

5. References

1. Backus, John. Can programming be liberated from the von Neumann style? A functional style and its algebra of programs. *Comm. ACM 21*, 8 (August 1978), 613-641.

2. Barendregt, H. P. *The Lambda Calculus.* North-Holland, Amsterdam, 1981, 59-62.

3. Burge, William H. *Recursive Programming Techniques.* Addison-Wesley, Reading, 1975.

4. Burstall, R., MacQueen, D., and Sannella, D. HOPE: An experimental applicative language. Internal Report CSR-62-80, Univ. of Edinburgh, 1980.

5. Church, A., and Rosser, J. Barkley. Some properties of conversion. *Trans. Amer. Math. Soc. 39*, (1936), 472-482.

6. Clocksin, W., and Mellish, C. *Programming in Prolog.* Springer Verlag, Berlin, 1981.

7. Curry, H. B., and Feys, R. *Combinatory Logic.* North-Holland, Amsterdam, 1958, Chapter 4.

8. van Emden, M. H., and Kowalski, R. A. The semantics of predicate logic as a programming language. *J. ACM 23*, 4 (October 1976), 733-742.

9. Glasner, I., Möncke, U., and Wilhelm, R. OPTRAN, a language for specification of program transformations. *Informatik-Fachberichte*, Springer Verlag, 1980, 125-142.

10. Henderson, Peter. *Functional Programming: Application and Implementation.* Prentice-Hall, Englewood-Cliffs, 1980.

11. Hindley, R. An abstract form of the Church-Rosser theorem, Part I. *J. Symbolic Logic 14* (1969), 545-560.

12. Hindley, R. An abstract Church-Rosser theorem, Part II. *J. Symbolic Logic 19* (1974), 1-21.

13. Hindley, R., Lercher, B., and Seldin, J. P. Introduction to combinatory logic. *London Math. Soc. Lecture Note Series 7*, Cambridge Univ. Press, 1972, Appendix 1.

14. Hoffman, C., and O'Donnell, M. J. Interpreter generation using tree pattern matching. *6th Annual Symposium on Principles of Programming Languages*, 1979, 169-179.

15. Hoffman, C., and O'Donnell, M. J. Pattern matching in trees. *J. ACM 29*, 1 (January 1982), 68-95.

16. Hoffman, C., and O'Donnell, M. J. Programming with equations. *ACM TOPLAS 4*, 1 (January 1982), 83-112.

17. Kowalski, R. Algorithm = Logic + Control. *C. ACM 22*, 7 (July 1979), 424-436.

18. Martin-Löf, P. An intuitionistic theory of types. unpublished manuscript, Univ. of Stockholm, 1972.

19. Newell, A. A theoretical exploration of mechanisms for coding the stimulus. in Melton, A. W., and Martin, E. (Eds). *Coding Processes in Human Memory*, Winston and Sons, Washington D.C., 1972, 373-434.

20. Newell, A., and Simon, H. A. *Human Problem Solving*. Prentice-Hall, Englewood-Cliffs, 1972.

21. Rosen, Barry K. Tree manipulation systems and Church-Rosser theorems. *J ACM 20*, 1 (January 1973), 160-187.

22. Rosser, J. Barkley. Highlights of the history of the lambda-calculus. *Conf Record of the 1982 ACM Symp. on LISP and Functional Prog.*, ACM Order No. 552820 (August 15-18, 1982), 216-225.

23. Schroer, D. E. The Church-Rosser Theorem. Ph.D. Thesis, Cornell U., Ithaca, 1965.

INITIAL DISTRIBUTION LIST

Defense Technical Information Center Cameron Station Alexandria, VA 22314	2
Dudley Knox Library Code 0142 Naval Postgraduate School Monterey, CA 93943	2
Office of Research Administration Code 012A Naval Postgraduate School Monterey, CA 93943	1
Associate Professor B. J. MacLennan, Code 52Ml Department of Computer Science Naval Postgraduate School Monterey, CA 93943	12
Chairman, Code 52Hq Department of Computer Science Naval Postgraduate School Monterey, CA 93943	40
Dr. Robert Grafton Code 433 Office of Naval Research 800 N. Quincy Arlington, VA 22217	1
Dr. David W. Mizell Office of Naval Research 1030 East Green Street Pasadena, CA 91106	1
Professor J. Barkley Rosser Mathematics Research Center University of Wisconsin - Madison Madison, WI 53706	1